

Training-Automate

Objetivo general

- Conectarnos a integrations y enviar información sobre los entrenamientos a esta plataforma centralizada.

Objetivos especificos

- Crear las apis que nos permitan conectarnos a **integrations**.
- Modificar los diccionarios de datos agregando unas claves que son necesarias para determinar la procedencia de quién enciende la máquina.

Actividades

El proyecto de **Training-Automate** fué modificado esencialmente en los siguientes archivos, pues la idea es no modificar aspectos básicos de lo que ya se tiene:

- **sending_ip.py**: Aquí se agregaron algunas variables como: **TOKEN_TRAINING** y **USUARIO_INTEGRATIONS**, además se agregaron dos funciones de consulta: **obtainTokenIntegrations** y **getWhoStartMachineIntegrations** al nuevo proyecto de aiting **Integrations**.

```
def obtainTokenIntegrations():
    """ Aqui se obtiene el token para acceder a las api """
    msg = {}
    headers = {
        "Manager": TOKEN_TRAINING
    }
    data = {
        "username": USUARIO_INTEGRATIONS,
    }
    url = f"{URL_BASE}/obtain-token/"
    try:
        response = requests.post(url, headers=headers, data=data)
    except Exception as e:
        response = None
        msg["error"] = str(e)
    finally:
        response = response

    if response and response.status_code == 200:
        msg["token"] = "Éxito"
        return response.json()["token"], msg
    else:
        return None, msg
```

```
def getWhoStartMachineIntegrations(token: str, name: str) -> str:
```

```

""" Aqui se obtiene la base de datos del py que enciende la máquina """
headers = {
    "Authorization": token
}
data = {
    "name": name,
}
url = f"{URL_BASE}/common/api/v1/who-start-machine/"
try:
    response = requests.post(url, headers=headers, data=data)
except Exception as e:
    response = None
    print(str(e))
finally:
    response = response

if response and response.status_code == 200:
    return response.json()['turn_on_by']
else:
    return None

```

- **training_dataset.py:** ubicada en la carpeta *views* del módulo **api** este archivo fué modificado agregando una nueva clave **turn_on_by** al diccionario esperado **data**, que posteriormente es enviado a una tarea celery.

```
turn_on_by = str(data["turn_on_by"])
```

- **training_dataset.py:** ubicada en la carpeta *task* del módulo **api** este archivo fué modificado agregando una nueva clave **turn_on_by** al diccionario esperado **data**, se agrego la función que nos permite autenticarnos ante **integrations** **obtainTokenIntegrations**, además se modificó el la url base del api a la cual apunta cuando finaliza el entrenamiento, paso de **saturno** a **integrations** y el el cual también viaja la misma clave **turn_on_by**.

```

response["turn_on_by"] = turn_on_by
BASE_URL = "https://integrations.aituring.co"
token, msg = obtainTokenIntegrations()
requests.post(
    f'{BASE_URL}/core/training/api/v1/finish_training/',
    data=response,
    headers={'Authorization': '{}'.format(token)}
)

```